Customizing Android's UI

Android Builders Summit 2015

Karim Yaghmour
@karimyaghmour

karim.yaghmour@opersys.com

>OPERSYS™

These slides are made available to you under a Creative Commons Share-Alike 3.0 license. The full terms of this license are here: https://creativecommons.org/licenses/by-sa/3.0/

Attribution requirements and misc., PLEASE READ:

- This slide must remain as-is in this specific location (slide #2), everything else you are free to change; including the logo :-)
- Use of figures in other documents must feature the below “Originals at” URL immediately under that figure and the below copyright notice where appropriate.
- You are free to fill in the “Delivered and/or customized by” space on the right as you see fit.
- You are FORBIDEN from using the default “About” slide as-is or any of its contents.
- You are FORBIDEN from using any content provided by 3rd parties without the EXPLICIT consent from those parties.

(C) Copyright 2013-2015, Opersys inc.

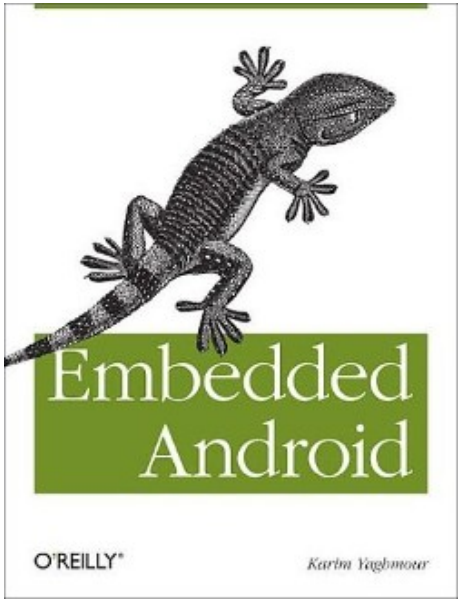
These slides created by: Karim Yaghmour

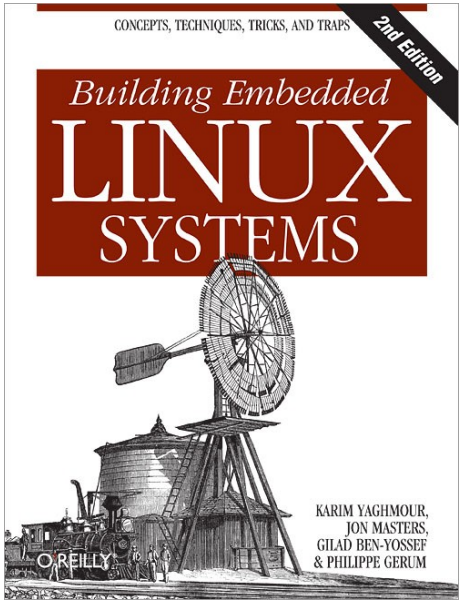
Originals at: www.opersys.com/community/docs

Delivered and/or customized by

About

- Author of:





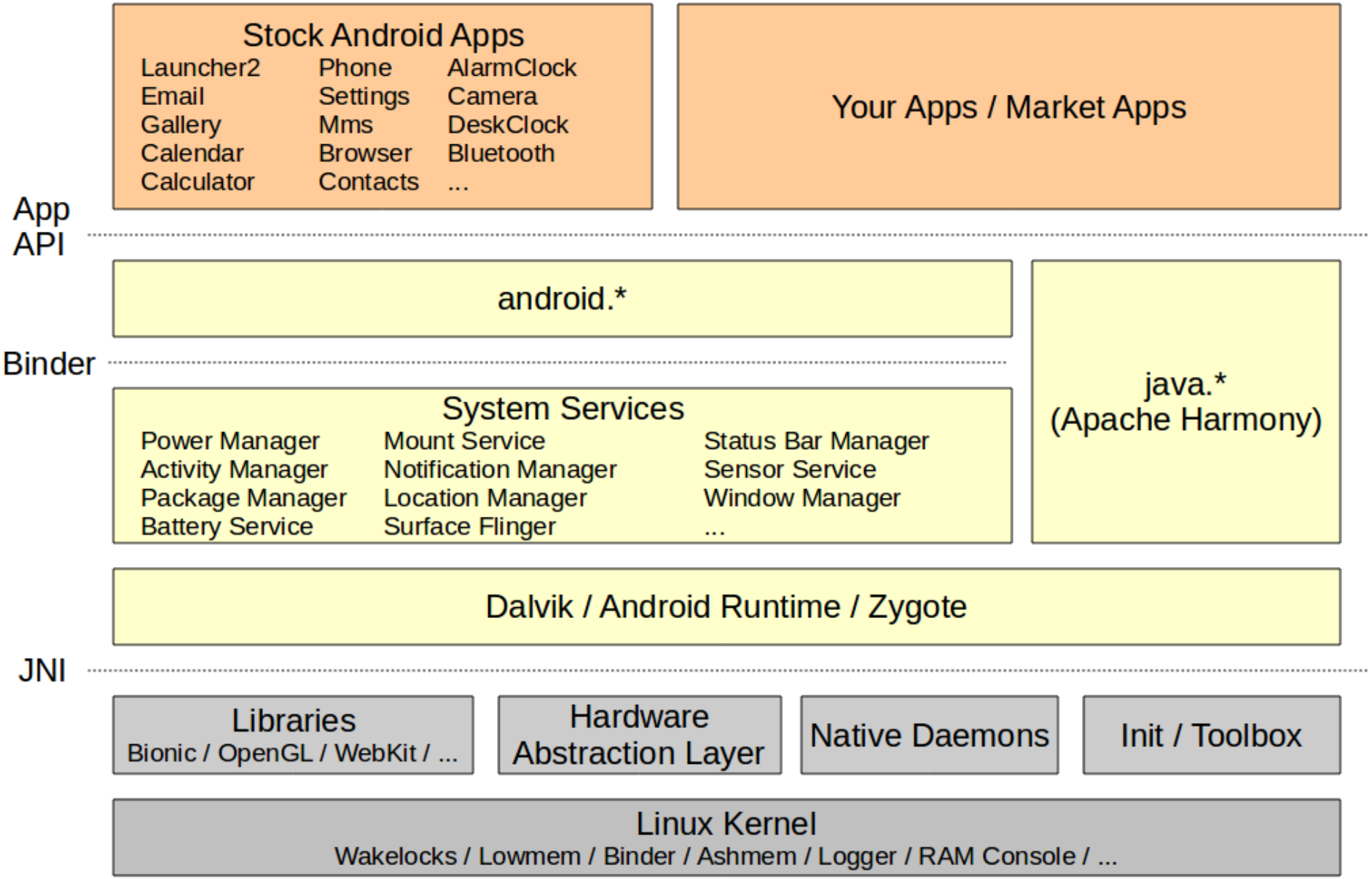
- Introduced Linux Trace Toolkit in 1999
- Originated Adeos and relayfs (kernel/relay.c)
- Ara Android Arch Oversight
- Training, Custom Dev, Consulting, ...

Agenda

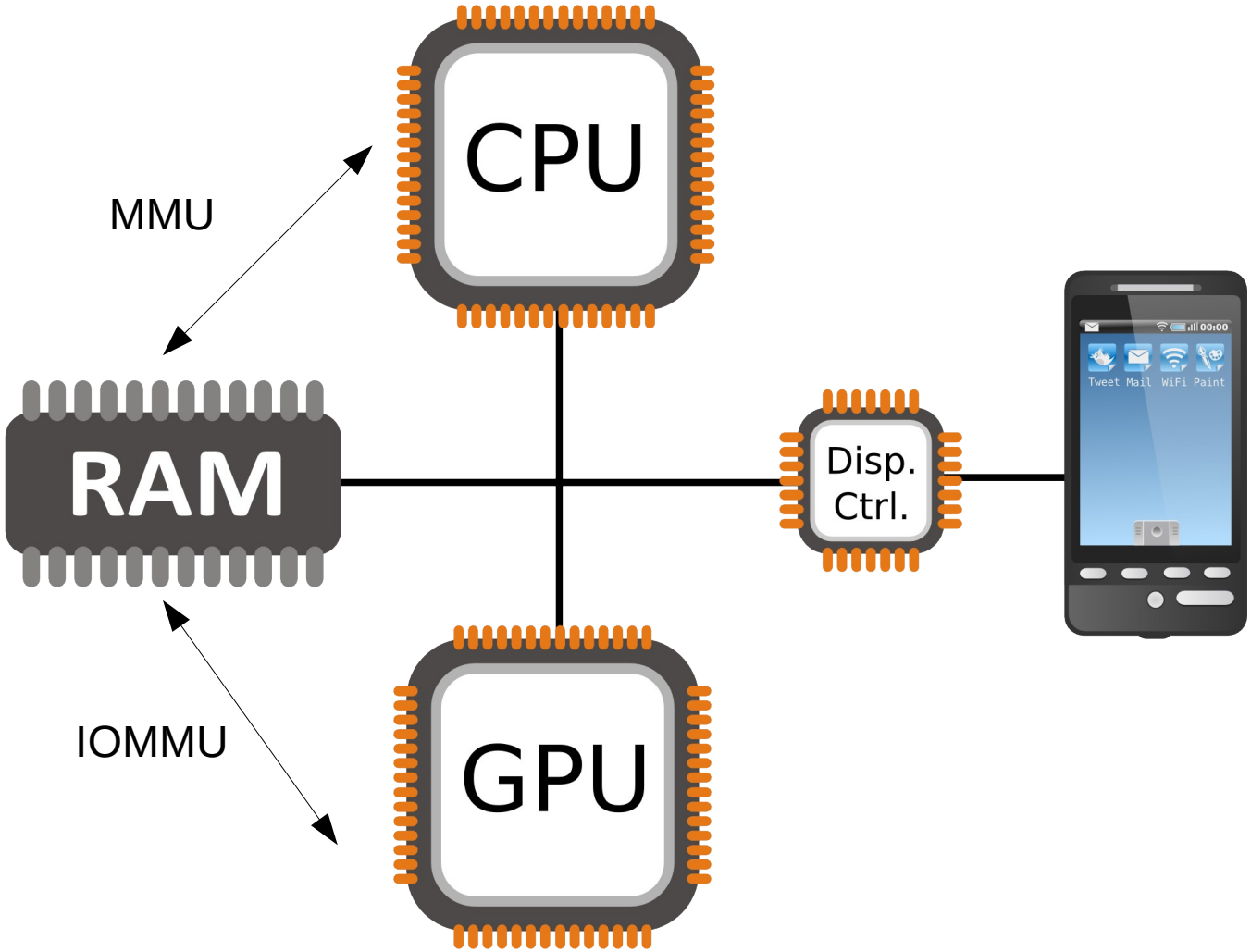
- Architecture
- Launcher App
- System UI
- Kiosk App
- Screen Overlays
- Theming
- Boot Animation
- Other UI Elements
- References

1. Architecture

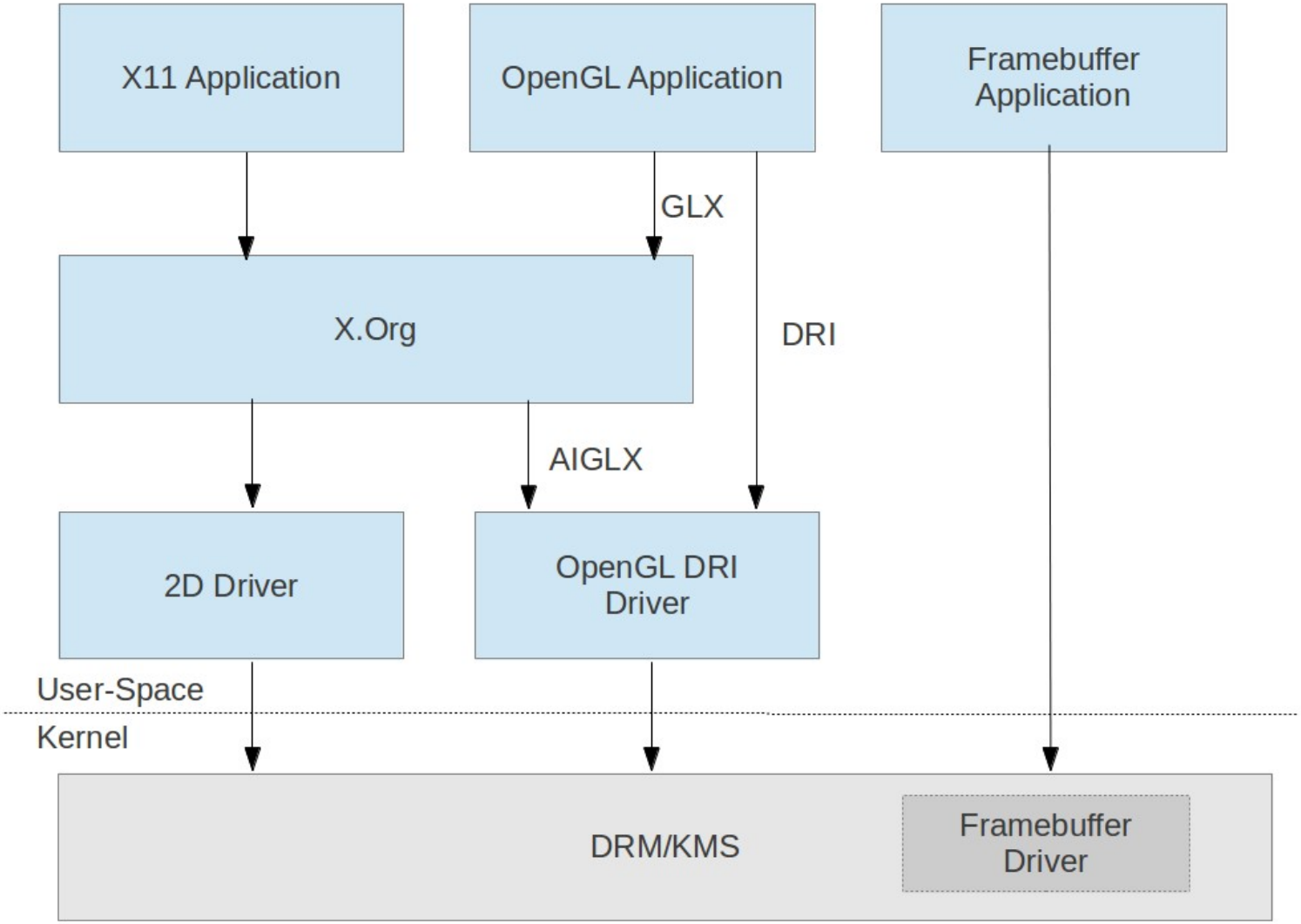
- Android Architecture
- Display Hardware
- Classic Linux Display Stack
- Display stack in Android
- HAL defininition
- HAL module
- Surface Flinger
- Window Manager


Stock Android Apps
Launcher2
Email
Gallery
Calendar
Calculator
Phone
Settings
Mms
Browser
Contacts
AlarmClock
Camera
DeskClock
Bluetooth
...
Your Apps / Market Apps
App API
android.*
Binder
System Services
Power Manager
Activity Manager
Package Manager
Battery Service
Mount Service
Notification Manager
Location Manager
Surface Flinger
Status Bar Manager
Sensor Service
Window Manager
...
java.*
(Apache Harmony)
Dalvik / Android Runtime / Zygote
JNI
Libraries
Bionic / OpenGL / WebKit / ...
Hardware Abstraction Layer
Native Daemons
Init / Toolbox
Linux Kernel
Wakelocks / Lowmem / Binder / Ashmem / Logger / RAM Console / ...

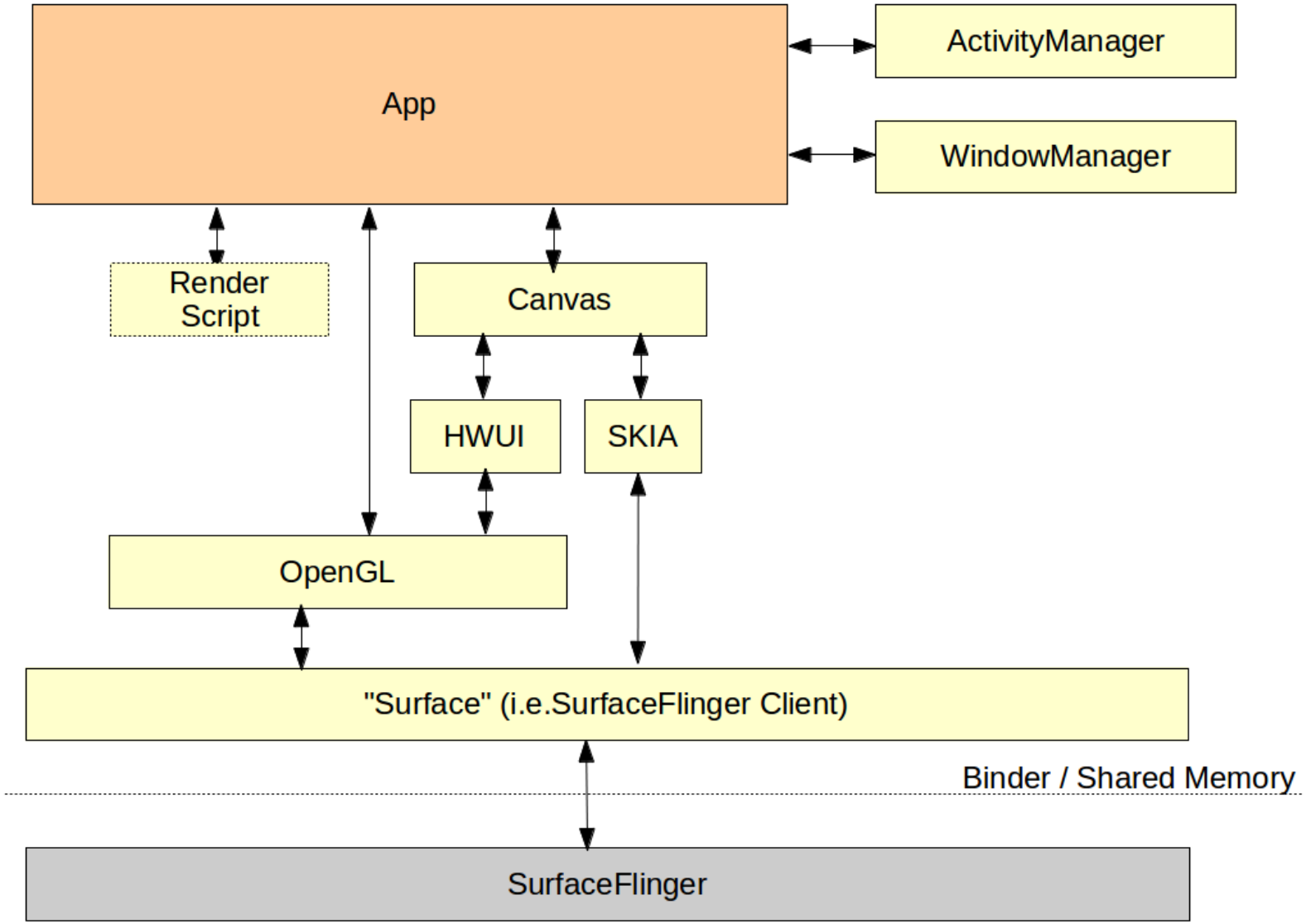
1.2. Display Hardware

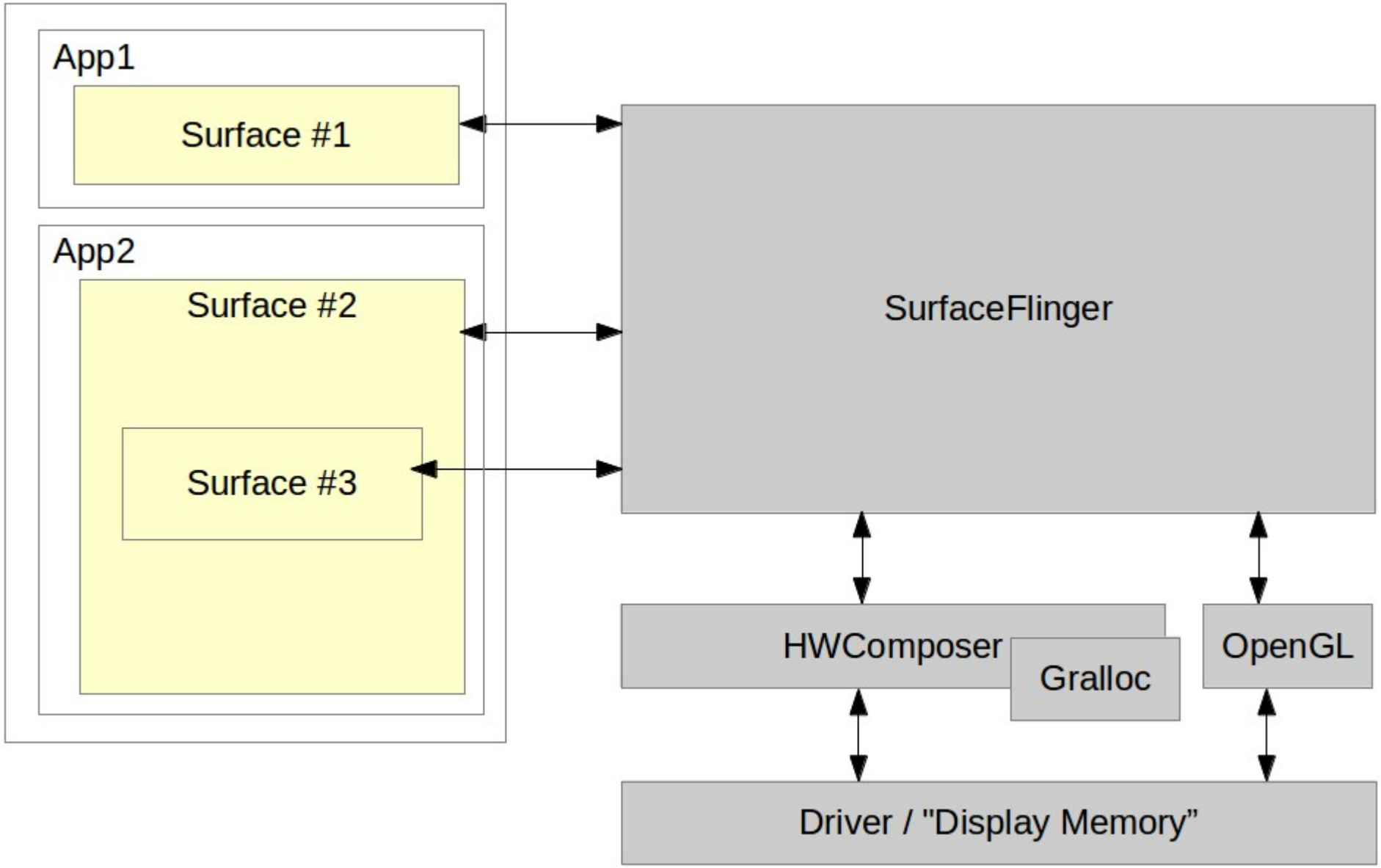


1.3. Classic Linux display stack



1.4. Display stack in Android




App1
Surface #1
App2
Surface #2
Surface #3
SurfaceFlinger
HWComposer
Gralloc
OpenGL
Driver / "Display Memory"

1.5. HAL Definition

- /hardware/libhardware/include/hardware/hwcomposer.h
- struct hwc_procs:
  - invalidate()
  - vsync()
- struct hwc_composer_device:
  - prepare()
  - set()
  - dump()
  - registerProcs()
  - query()
  - *()

1.6. HAL module

- Skeleton /hardware/libhardware/modules/hwcomposer.cpp
- /system/lib/hw/hwcomposer.BOARD.so
- /system/lib/hw/gralloc.BOARD.so
- Ex. - Mot Xoom:
  - hwcomposer.tegra.so
  - gralloc.tegra.so
- Surface Flinger hook:
  - /frameworks/native/services/surfaceflinger/DisplayHardware
    - HWComposer.cpp
    - Provides fake vsync if none is provided in HW

1.7. Surface Flinger

- Actual server:
  - /frameworks/native/services/surfaceflinger
- Client side:
  - /frameworks/native/libs/gui
- Client / Server interface:
  - ISurfaceComposerClient.cpp
  - ISurfaceComposer.cpp
- This is NOT an aidl'ed service
- All communication is manually marshalled/unmarshalled

1.8. Window Manager

- Server side:
  - /frameworks/base/services/java/com/android/server/wm/
    - WindowManagerService.java
    - Session.java
- Client side:
  - /frameworks/base/core/java/android/view/
    - WindowManager.java
    - WindowManagerImpl.java
    - ViewRootImpl.java
- Interfaces:
  - IWindowManager.aidl
  - IWindowSession.aidl
- Parameters (incl. z-order):
  - See WindowManager.java

2. Launcher

- This the “Home” App
- Compiled as part of AOSP
- See /packages/app/Launcher2
- There's actually a real “Home” App:
  - development/samples/Home
- Android.mk:

```
LOCAL_OVERRIDES_PACKAGES := Home
```


>OPERSYS™

- AndroidManifest.xml:

```
    <activity
...
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.HOME" />
...
```

- At least two ways for it to be triggered:
  - Activity Manager sends android.intent.category.HOME intent at startup
  - Clicking on the “Home” button
- What this app DOESN'T do:
  - Use any of the hidden APIs (LOCAL_SDK_VERSION := current)
  - Be signed by the platform key (LOCAL_CERTIFICATE := shared)
  - Be persistent

3. System UI

- Comprises:
  - Status Bar
  - Navigation Bar
  - Screen Shot
  - Load Average Service
  - Wallpaper
  - Recent Activities
  - USB settings
- A unique app
- See /frameworks/base/packages/SystemUI
- Connects to Status Bar Manager and gives an interface it can use to call back into Status Bar

4. Kiosk App

- Just replace the Launcher/Home with your home
- Implement an activity that catches:

  android.intent.category.HOME

- You may want to disable the status bar

5. Screen Overlays

- You can talk to the Window Manager directly
- Have it add a View above the current display
- Example:
  - Load Average service in SystemUI

6. Theming

- Nothing really different from regular resources
- More art than programming
- See frameworks/base/core/res/res
- Generates /system/framework/framework-res.apk
- Lots of theming mods out there show how to manipulate this apk.
- But since you've got the sources ...
- Examples:
  - https://github.com/elelinux/HTC.Elegance
  - https://github.com/nadeemsultan/HolofiedCM11Theme

- More resources on Theming:

  http://xda-university.com/as-a-user/introduction-to-basic-rom-theming

  http://janrain.com/blog/introduction-to-android-theme-customization/

  http://www.vogella.com/tutorials/AndroidStylesThemes/article.html

  http://www.phonesdevelopers.com/1717751/

7. Boot animation

- Started by Surface Flinger
- “bootanim” binary
- /frameworks/base/cmds/bootanimation
- Relies on bootanimation.zip w/ PNGs (nothing but)
- See https://github.com/CyanogenMod/android_vendor_cm/tree/jellybean/prebuilt/common/bootanimation
- Must contain a desc.txt:

  <width> <height> <fps>

  p <count> <pause> <path>

  p <count> <pause> <path>

>OPERSYS™

8. Other UI Elements

- Wallpaper Manager Service
- Notification Service
- App Widgets

8.1. Wallpaper Manager Service

- See /frameworks/base/services/java/com/android/server/WallpaperManagerService.java

>OPERSYS™

8.2. Notification Service

- Toasts
- Status bar notifications
- Gets handle to Status Bar Service at instantiation
- Uses handle to communicate with Status Bar

8.3. App Widgets

- See /frameworks/base/services/java/com/android/server/AppWidgetService.java

Thank you ...

karim.yaghmour@opersys.com

>OPERSYS™